

## REMARKS/ARGUMENTS

Claims 1-39 are pending in the present application. Claims 1-39 were canceled and claim 40 was added. Reconsideration of the claims is respectfully requested.

### **I. 35 U.S.C. § 103, Obviousness**

The Examiner has rejected claims 1-39 under 35 U.S.C. § 103 as being unpatentable over U. S. Patent No. 5,418,950 (hereinafter Li) and further in view of U. S. Patent Publication No. 2003/0167258. New claim 40 is directed to more particularly point out Applicants' invention. The need that the method of claim 40 meets is to automatically form a database query from a template while only joining necessary tables. The template, by itself, would join all tables in the database to locate requested information.

receiving a template which contains a necessary language for the database query and a plurality of placeholders but lacks a plurality of clauses required for the database query to produce a desired output;	[0029]
receiving a user entry of the plurality of fields, the plurality of filters, and a sort criteria;	[0010], line 4
responsive to receiving the user entry of the plurality of fields, the plurality of filters, and the sort criteria, retrieving a plurality of aliases for the plurality of fields and the plurality of filters from a column map of a database;	[0019], [0034], line 8
generating a SELECT clause within the plurality of clauses based on the plurality of fields;	[0035], lines 8-12
generating an ORDER BY clause within the plurality of clauses based on the sort criteria;	[0037], [0025]
generating a first portion of a WHERE clause within the plurality of clauses based on the plurality of filters;	[0036] ("(filter) WHERE clause" is "first portion")
creating a first list, which is initially empty;	[0038], line 4 ("added alias list" is "first list"
creating a second list, which is initially empty;	[0039], line 2 ("optional WHERE clause alias list" is "second list"
determining whether each of the plurality of aliases is present in the first list;	[0038], line 3-4
responsive to a determination a first alias from the plurality of aliases is not present in the first list, performing steps comprising: when the template contains a FROM placeholder among the plurality of placeholders, inserting the first alias with a matching table name into a FROM clause within the plurality of clauses;	[0042], line 7-9
when the template does not contain the FROM	[0043], lines 1-4, 10-12,

placeholder among the plurality of placeholders but contains a JOIN placeholder among the plurality of placeholders, inserting the first alias with a matching table name into a JOIN clause within the plurality of clauses;	
when the template contains neither the FROM placeholder among the plurality of placeholders nor the JOIN placeholder among the plurality of placeholders, returning an error;	[0044], lines 2-4
generating a second portion of the WHERE clause based on a joining condition for the first alias;	[0044], line 5 (“(join) WHERE” is “second portion”
joining the first portion and the second portion of the WHERE clause;	[0040], lines 9-10
adding the first alias to the first list;	[0038], lines 3-6, [0044], lines 11-12
adding the first alias and the second portion of the WHERE clause to the second list, when the second portion of the WHERE clause depends on a second alias from the plurality of aliases;	[0044], lines 12-13 (“optional WHERE clause aliases list” is “second list”
determining whether each of the plurality of aliases in the second list is present in the first list;	[0039], lines 7-8
responsive to a determination that a third alias is present in both the second list and the first list, adding the third alias to the second portion of the WHERE clause;	[0039], lines 2-5
replacing the plurality of placeholders with the plurality of clauses;	[0040], lines 3-5
wherein the database query joins only a plurality of tables from the database necessary to complete the query only when those aliases found in the first list and in the second list were added to the second portion of the WHERE clause.	[0039], lines 8-12

The invention takes in the user’s desired fields (columns), filters (e.g. order\_id = 1234) and a sort order, generates a query that joins only tables necessary to complete the query, and runs the query. The user need not specify the FROM or JOIN clauses of the query (i.e. which tables to join) in order to get the results because the invention generates those clauses based on the user’s inputs.

The query is generated by first making a list of “parameters” out of the fields and filters, which involves looking up an alias for the columns requested by the fields and filters in a column map of various tables. Parameters are individually checked against an “added alias list,” which starts out empty. In the event a parameter is not on the “added alias list,” its alias is added to the list and added to the appropriate selecting or filtering clauses(s) in the query. All aliases have an associated “optional where clause” that performs joining, and at this point the invention looks for any aliases whose “optional where clause” depends on another alias. Aliases that have such a dependency are added to an “optional where clause

alias list.” At the end of analyzing all parameters, the “optional where clause list” and the “added alias list” are compared, and only those “optional where clauses” that appear in both lists are added to the WHERE clause of the SQL statement. It is the comparison of the lists and adding only those optional where clauses that appear in both lists to the WHERE clause of the SQL statement that ensures that only the necessary joining statements are added to the query.

Claim 40 is presented to clarify the operation of Applicants’ invention. For example, the (filter) WHERE clause is referred to as the “first portion” of the WHERE clause and the (join) WHERE clause is referred to as the “second portion” of the WHERE clause. The “added alias list” is the first list, and the “optional WHERE clause aliases list” is the “second list.” Applicants submit that claim 40’s elements cooperate to produce a novel result, and the novel result is set forth in the final limitation of claim 40 which states “wherein the database query joins only a plurality of tables from the database necessary to complete the query only when those aliases found in the first list and in the second list were added to the second portion of the WHERE clause.” In other words, none of the prior art references, individually or in combination teach or suggest preventing the joining of unnecessary tables in the manner specified by claim 40.

Neither Li nor Koo disclose or suggest the comparison of the lists and adding only those optional where clauses that appear in both lists to the WHERE clause of the SQL statement that ensures that only the necessary joining statements are added to the query. Moreover, Li generates its own entire queries, and does not disclose making determination based on a pre-existing SQL template. Li does not fill in SQL templates; rather Li generates entire queries using a graphical user interface.

Therefore, claim 40 distinguishes over the art of record and is in condition for allowance.

## II. Conclusion

It is respectfully urged that the subject application is patentable over Li and Koo and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: October 15, 2008

Respectfully submitted,

/Rudolf O. Siegesmund/

Rudolf O. Siegesmund  
Reg. No. 37,720  
Yee & Associates, P.C.  
P.O. Box 802333  
Dallas, TX 75380  
(972) 385-8777  
Attorney for Applicants